Seneca

Assignment 1: Effective use of Terraform to deploy multi-environment infrastructure

Submission Instructions	To be submitted via Blackboard. Refer to Blackboard for submission instructions
Value	15% of final grade
Due Date	

Learning Outcomes Covered in Assignment
1. Explain benefits of DevOps practices and deployment automation to support organizational shift towards DevOps culture 2. Analyze application's functional and operational requirements to recommend software development lifecycle components 3. Apply Infrastructure as a Code approach to deploy all parts of the application hosting solution in a repeatable and reliable way 4. Support the security posture of the cloud infrastructure by identifying and implementing preventive and detective security controls

Table of Contents

1. Assignment Outline

The objective of this assignment is to verify your skills in applying you Terraform skills to deploy infrastructure outlined in the design diagram below.

In addition to your Terraform skills, the assignment verifies your understanding of modularized approach to Terraform configuration, distributed state, use of functions, conditions, and loops, and the correct use of naming and tagging conventions.

Moreover, the assignment tests your understanding of networking components used to create secure cloud solutions with a high level of segregation between production and non-production n environments. The deployed resources should adhere to the clearly defined naming convention and should have meaningful tags applied to all the resources (where applicable).

As a result of this activity, you will create a Terraform configuration that deploys 2 VPCs that are simulating preproduction and production environments. The VPCs will be connected via VPC peering. The EC2 instances deployed into the respective VPCs will allow limited ingress and protocols and ports. The detailed diagram is available in Section 2, Architecture.

The Terraform configuration should be modular and should support environment-specific deployment.

Each completed task should be supported by the artifacts outlined in the submission requirements and Appendix.

2. Architecture

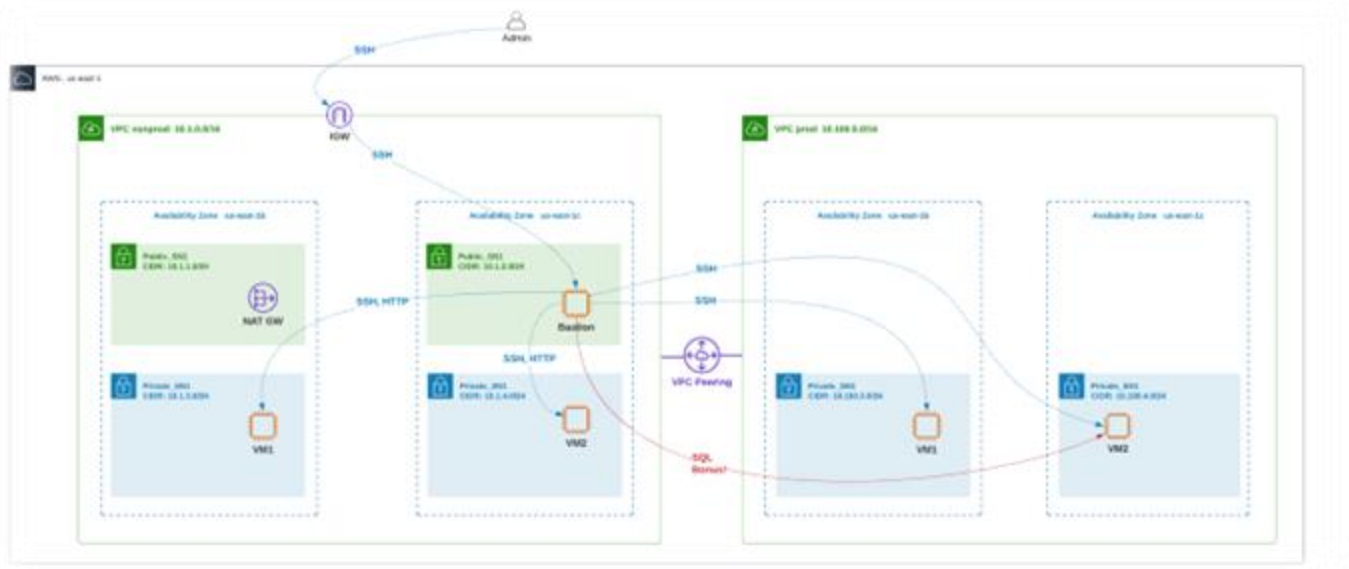

Diagram explained:

- VPC nonprod has 2 private and 2 public subnets. Bastion host and NAT gateway are deployed into public subnets in nonprod VPC. VM1 and VM2 are deployed in private subnets of nonprod VPC. Apache webserver is deployed on VM1 and VM2 in nonprod VPC.
- VPC prod has 2 private subnets and no public subnets. VM1 and VM2 are deployed into private subnets of prod VPC.
- VPC prod and VPC nonprod are connected via peering connection.
- Admins can connect to bastion host via SSH using ec2-user. Admins can connect from bastion host to all 4 VMs in prod and nonprod VPCs via SSH using ec2-user. Admins can send HTTP requests to Apache webservers running on VM1 and VM2 in nonprod VPC.

3. Submission

Your submission should include 2 separate files:

1. Zip file with the Terraform code that provisions all the infrastructure defined in Section 2, Architecture diagram. The submitted code **should not include** any hidden folders (e.g. .terraform, .c9) or ssh keys. The submission has to include README file that clearly states the deployment steps. The README file should be in the root folder. For more details, please see Section 5, Submission Requirements Description.

2. Recorded assignment presentation. Please see Section 8 with the link to the assignment presentation recording.

   Important Note: Make sure that all the network connectivity rules have been implemented with the principle of least privilege in mind.

4. Submission Requirements Description

4.1. Terraform Code

Task	Submission Requirements Description
README	The README file should clearly explain all the pre-requisites for the successful deployment (e.g. S3 bucket that stores Terraform state and an SSH key should be pre-created). README should provide instructions for successful deployment and cleanup of the provided solution.
Infrastructure deployment automation: Deployment of VPCs, public and private subnets, route tables, NAT GW, Internet GWs, peering connection Security groups definition EC2 instances deployment	The Terraform configuration should be parametrized and modularized. Avoid code repetitions. Make sure your code supports different type and number of VMs in prod and nonprod environments. The solution should use remote Terraform state with the segregated networking and VMs components. Ensure your Security Groups with the principle of least privilege in mind.
Configuration of EC2 instances: VM1 and VM2 in nonprod VPC should have httpd installed and configured to serve a custom message. with its IP, your name and the name of the environment.	VMs in nonprod environment should serve a web traffic with a custom message that prints out your name, the name of the environment and the private IP address of the EC2 instance. VMs in prod environment should not have any additional packages installed.

VM1 and VM2 in prod VPC should not have any extra packages installed. Bastion host should not have any extra packages installed.	Bastion host in dev environment should not have any additional packages installed.
Bonus! Install MySQL client on bastion host in nonprod VPC.	Configure all the networking components to allow MySQL client to send SQL queries to VM2 in prod VPC.

4.2. Recording

Task #	Submission Requirements Description
	Introduction Describe in your own words the main objectives of this assignment
1	Demonstrate that admins can access bastion host via SSH and can connect to the EC2 instances in private subnets in **nonprod** VPC via SSH.
2	Demonstrate that admins can access bastion host via SSH and can connect to the EC2 instances in private subnets in **prod** VPC via SSH.
3	Demonstrate that admins can send HTTP requests to Apache webservers running on VM1 and VM2 in nonprod VPC.
4	Explain the pros and cons of network connectivity between prod and non-prod environments.
5	**Bonus**! Demonstrate that SQL client running on bastion host has network connectivity to VM2 in prod VPC.
	Conclusion

5. Plagiarism:

Plagiarized assignments will receive a mark of zero on the assignment and a failing grade on the course. You may also receive a permanent note of plagiarism on your academic record.

6. Assignment Grade Breakdown

Section/Task		Points
	Introduction	3
	1. Demonstrate that admins can access bastion host via SSH and can connect to the EC2 instances in private subnets in **nonprod** VPC via SSH.	5
	2. Demonstrate that admins can access bastion host via SSH and can connect to the EC2 instances in private subnets in **prod** VPC via SSH.	5
	3. Demonstrate that admins can send HTTP requests to Apache webservers running on VM1 and VM2 in nonprod VPC.	5
	4. Explain the pros and cons of network connectivity between prod and non-prod environments.	5
	Bonus! 5. Demonstrate that sql client is running on bastion host.	5
	Cleanup	5
	Conclusion	5
	README	7
	Use of remote Terraform state	
	Infrastructure deployment automation:	10

Terraform code	Deployment of VPCs, public and private subnets, route tables, NAT GWs, and Internet GWs	
	Infrastructure deployment automation: Security groups definition	10
	Infrastructure deployment automation: VPC Peering **Note**: Create VPC peering manually through AWS Management console if Terraform implementation is too challenging	15
	Infrastructure deployment automation: Route tables definitions to allow peering connection **Note**: Update route tables manually through AWS Management console if Terraform implementation is too challenging	10
	Configuration of VMs automation (static web site and administration tools on bastion host)	10
	Bonus! Configuration of MySQL client	5
	Total	**100 + 10 Bonus points**

Important Notes:

- The code that cannot be deployed based on the instructions provided in the README **will result in grade 0 for the Terraform Code section.**

- The Terraform configuration should be modular, should use functions, conditions, input, and output variables, data sources and other advanced Terraform features we learned over the past 3 weeks. The solution should implement multi-environment folder structure. If you struggle to implement some part of infrastructure with terraform and did it manually through console, **please state it explicitly.**

- **Cleanup of all the deployed infrastructure is crucial in this assignment.** NAT GW cost ~140USD/month and will consume all the budget allocated to this course if left running.

7. Suggested Implementation Steps

Start with the code created in `week5/Session2/02_task2_completed`.

1. Extend the network module to support private and public subnets in dev VPC. Deploy dev VPC.
2. Refactor the network module to support the design of dev and prod VPCs. Pay attention to gateways and routing rules. Deploy prod VPC.
3. Create another local Terraform module that establishes peering connection between VPC. Look at the remote modules implementation by cloudposse but create your own Terraform module.
4. Deploy peering connection between VPCs, locate peering configuration at the same level as dev and prod folders.
5. Deploy webservers and bastion host in dev VPC
6. Deploy vanilla Linux VMs in prod VPC
7. Verify connectivity as per Section 4.2

8. Appendix – Recording Requirements

The submission of this project should be done via recording.
The recording should be no more than 15 min long. The recording should cover the points below:

1. Deployment of all the infrastructure
    a. VPCs, subnets, route tables and routes. IGW and NAT GW in dev and prod
    b. Creating peering connection
    c. Creating EC2 instances in dev and prod
2. Required functionality based on Section 4.2
    a. Demonstrate that admins can access bastion host via SSH and can connect to the EC2 instances in private subnet in **nonprod** VPC via SSH.
    b. Demonstrate that admins can access bastion host via SSH and can connect to the EC2 instances in private subnets in **prod** VPC via SSH.
    c. Demonstrate that admins can send HTTP requests to Apache webservers running on VM1 and VM2 in nonprod VPC.
    d. Explain the pros and cons of network connectivity between prod and non-prod environments.
3. Bonus! Modify your deployment to provide the bastion host with the ability to run SQL queries against VM2 with in prod VPC. No need to install MySQL server on VM2 in prod VPC.
4. Conclusion with an outline of the issues you faced and resolved during the assignment and the topics that you mustered.